Telecommun Syst (2013) 52:1021–1031
DOI 10.1007/s11235-011-9609-y

Dynamic host configuration protocol for IPv6 improvements for mobile nodes

Tomasz Mrugalski · Jozef Wozniak · Krzysztof Nowicki

Published online: 6 October 2011

Abstract In wireless networks mobile clients change their physical location, which results in changing point of attachment to the network. Such handovers introduce unwanted periods, when node does not have communication capabilities. Depending on many conditions, such events may require reconfiguration of layer 2 (e.g. IEEE 802.16) or both 2 and 3 layers (IPv6). This paper investigates delays introduced in the latter type of handover. IPv6 protocol family supports two automatic configuration modes: stateless (SLAAC) and stateful (DHCPv6). Both modes may be used in wireless networks. Once the L2 handover procedure is completed, the mobile node (MN) starts its IPv6 configuration process, using stateless (router advertisements) or stateful (DHCPv6) mode. When care-of address (CoA) is assigned, its uniqueness has to be verified, using Duplicate Address Detection (DAD) procedure. Depending on a network type, this procedure may even take more than 1000 ms. The obtained CoA can be used only when configuration and DAD procedures are completed for informing corresponding nodes about new MN location. Such delay introduces unacceptable gaps in communication capability. This paper proposes several new mechanisms that enable faster IPv6 reconfiguration. First proposal allow MN to obtain its IPv6 address and other configuration options in advance, before completing actual handover. Such a priori knowledge about configuration available at destination locations may be exploited to speed up configuration process itself and also allow initiating Mobile IPv6 operations earlier, thus further shortening delays. Another proposal includes new way of delivering routing information to MN, using DHCPv6. Mechanism itself and its verification techniques are discussed. Results of extensive simulations, statistical analysis as well as areas of further study conclude this paper.

Keywords Autoconfiguration · DHCPv6 · DAD · Mobile IPv6 · Remote autoconfiguration · Routing · WiMAX

T. Mrugalski (✉) · J. Wozniak · K. Nowicki
Gdansk University of Technology, ul. Narutowicza 11/12, 80-233 Gdansk, Poland
e-mail: tomasz.mrugalski@eti.pg.gda.pl

J. Wozniak
e-mail: jowoz@eti.pg.gda.pl

K. Nowicki
e-mail: know@eti.pg.gda.pl

1 Introduction

With wireless technologies reaching their maturity, more users are expected to use mobile devices. At the same time, the growing speed of data transfers, offered by wireless networks causes that multimedia applications, like VoD (Video on Demand) or VoIP (Voice over IP), are becoming increasingly popular among mobile users. Wireless devices require service continuity even when they move between points of attachment. Thus the handover performance becomes a crucial factor for multimedia services support. These types of services are very sensitive to the channel disruption, handover delays or packet losses. All these factors can significantly lower the quality of multimedia services. Due to this, it is not possible to support multimedia without fast enough and transparent handover procedures.

However, from the network point of view, two requirements—delivery of large amounts of data and provision of mobility—are very hard to be achieved at the same time. That is because changing a point of attachment to the network by a mobile station is usually complicated.

Even though discussed problems are applicable to most wireless networks supporting inter-domain handovers, authors choose IEEE 802.16 networks (WiMAX, [5]) as a suitable research area, with the intent to attempt generalization to other networks. Different network layers will produce dramatically different delays during handover. The PHY and MAC layers of the WiMAX stack have been developed with mobility support and fast processing in mind. Therefore delays introduced are considered small (reaching a few hundred milliseconds range, usually slightly above 100 ms). Unfortunately, IPv6 protocols family was not designed in this manner. Several steps necessary to be performed by mobile nodes, while changing the network domain, introduce delays that are very large from the mobility point of view (in the order of one second or more). For example, the DHCPv6 server discovery phase takes exactly one second as clients are required to wait for possible responses from other servers, even when one or more servers have already responded (according to DHCPv6 specification [4]).

2 Problem statement

It is essential to realize that not all handover steps are causing handover delays. From the user's perspective, only lack of communication capability periods are troublesome. Therefore all efforts presented in this paper are focused on minimizing or even eliminating such periods completely. IPv6 reconfiguration process during handovers in wireless networks in general and IEEE 802.16 networks in particular, is not optimal and it is possible to achieve improved handover efficiency by performing remote DHCPv6, DAD and Mobile IPv6 protocol modifications. To measure impact of diverse algorithms in a uniform way, a new metric has been defined for assessment purposes.

Inter-domain handover in IPv6/WiMAX networks is time consuming and complicated process. During certain steps, like scanning or IPv6 autoconfiguration, a subscriber station is unable to maintain communication. To conveniently assess and compare radically different handover phases, metric called Handover Delay is proposed. It is expressed in milliseconds and specifies how long mobile IPv6 station does not have full communication capability due to the analyzed method. X expresses metric value, while HD() stands for its symbolic designation:

$$X = \mathrm{HD}(\mathrm{step})\ (\mathrm{ms})$$

In general, lower scored methods are considered "better", because they introduce smaller latency. If a method allows IPv6 node to communicate immediately, with no delay at all, its HD value is equal to 0 ms, thus it does not hinder communication in any way, so—assuming no other dependencies—it does not require any optimizations or improvements. Handover Latency would be a better term to describe this property as during handover packets (or traffic in general) are not delayed. However, this metric is more often referred to using its colloquial name—Handover Delay [15].

3 Previous work

Inter-domain handover optimization is an area of very active studies. One area of particular importance are activities arranged around IETF. Due to work fragmentation and varied approaches, there are many IETF working groups (WG) that are dedicated to solve various aspects of handover and mobility in general. Notable WGs are: mip4 (dedicated to Mobile IPv4 protocol development), mip6 (concluded; dedicated to Mobile IPv6 specification), mipshop (Mobile IP Performance, Signaling and Handoff Optimization, focused on Fast Handovers and Hierarchical extensions to Mobile IPv6), dna (Detecting Network Attachment; created to develop mechanisms that reduce or avoid delays associated with RA and DAD mechanisms), mext (Mobility Extensions for IPv6, like Network Mobility or IKEv2 usage). The most widely accepted extensions to Mobile IPv6 protocol are Hierarchical Mobile IPv6 (hmipv6) [18] and Fast Handovers for Mobile IPv6 (fmipv6) [6].

Fast Handovers for Mobile IPv6 [6], released in July 2009, is a set of procedures dedicated to improve handover latency. Previous Access Router (PAR) and New Access Router (NAR) are introduced. Using Proxy Router Advertisements (PrRA), it is possible that MN learns prefixes available at potential destination locations. MN communicated with routers PAR and NAR using Fast Binding Update (FBU) and Fast Binding Acknowledge (FBA) messages. By using Handover Indication (HI) and Handover Acknowledge (HA), PAR and NAR can coordinate traffic buffering and forwarding. Signaling for handover completion is also introduced. Two modes of operation are introduced: predictive and reactive. Large number of new messages requires significant modification of the protocol implementations. Also, the need to deploy mobility aware routers that in some cases need to buffer incoming traffic are cause of significant scalability and deployment concerns. Also, MN may obtain address for destination location using PrRA that must be confirmed using another message. This approach violates clear distinction between stateless and stateful autoconfiguration modes.

Second important work is HMIPv6 standard [18]. Mobility Anchor Point (MAP), a central router handling all traffic to and from a domain, is defined. MN arriving at new domain, registers MAP's address (called Regional Care-of Address, RCoA) to its HA, but also registers its locally obtained address (Local Care-of Address, LCoA) to MAP. By providing two levels of indirection, user traffic needs to be

processed by MAP (which serves as a HA-MAP tunnel termination point). This two level registration allows significant optimization, however. Mobility within a domain can be reported to MAP. As it is in the same domain, RTT times are much shorter, so expected handover delay is much shorter.

Other interesting proposals in related areas are Optimistic DAD [10] that leverages the assumption that address duplicates are extremely unlikely. Using modified ICMP Redirect messages, newly obtained addresses may be used immediately, before DAD procedure completes. Scope of usage of addresses used in this mode has certain restrictions, however.

Another important development in the area of mobility are Media Independent Handover services, published as IEEE 802.21 specification [1]. It introduces set of functions and notifications that different layers of protocols stack may use to gather and provide information regarding handover state to other layers. Event, Command and Information services are defined. By leveraging such information, it is possible to leverage existing indicators to optimize certain handover procedures, e.g. prepare for imminent handover due to degrading signal quality.

4 Remote autoconfiguration using DHCPv6

During normal handover procedure, data link layer (e.g. 802.16) initiates and performs handover procedure. This phase is often referred to as L2 handover. After such procedure is completed, network layer (e.g. IPv6) handover is performed. Doing so, delays introduced by each layer are adding up, resulting in a large overall delay. Using data gathered by IEEE 802.16, subscriber knows its target location, before actual handover occurs. This prior knowledge may be exploited to initiate connection with a DHCPv6 server, located at the destination network. As all base stations are connected to the Core Network (CN), it is possible to make connection between base stations using CN. To initiate and maintain such communication, already existing DHCPv6 relays may be used, albeit in a modified form. In a classical configuration, relays work as intermediaries between clients and servers. From the client's perspective, direct communication with a server or via relays is indistinguishable. Relays act as representatives of the server. From the server's perspective client is connected to the remote link. By modifying relay's behavior, it is possible to use relays to forward data from a client to a server and vice versa. In this scenario, the client is aware of the relays. It sends messages to relays and expects them to be forwarded to the remote server. Thus relays act as representative of the client. From the server's perspective, client is connected directly to the local link. To achieve such operation, relays and servers must

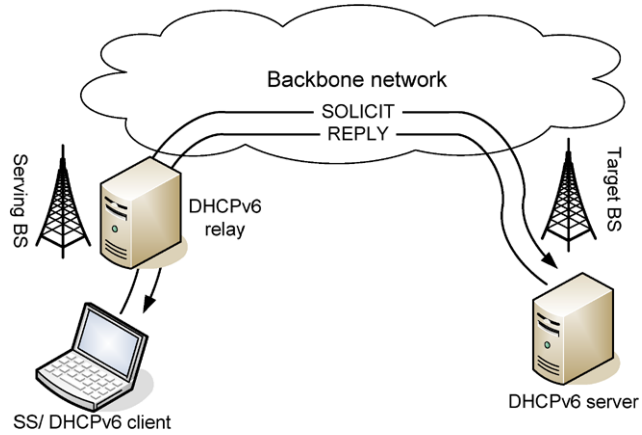

Fig. 1 Remote autoconfiguration of the mobile IPv6 nodes. Mobile Node communicates with its target location, while still maintaining full connectivity at the old location. Communication is achieved via Core Network

support this new mode. It is client's responsibility to define, which destination server it would like send DHCPv6 requests to.

Client should include extra option to indicate, which server it would like connect to. This information will be used by relays to forward messages to final destination. Knowledge about destination location identifiers is provided by WiMAX layers. "BS ID" obtained during scanning and/or L2 handover preparation is a functional equivalent of the MAC address. DHCPv6 server unique identifier (see [4]) of type DUID-LL (DHCPv6 Unique Identifier, based on link address) can be generated from such information. It is up to the relay to find actual location of the destination server. Overview of this improvement is depicted on Fig. 1.

This approach allows mobile node to communicate with target location before actually changing point of attachment to the network. This ability can be used to remotely obtain all required configuration parameters, including IPv6 address. Such a priori knowledge about target location can be used in several ways. All parameters may be used immediately after reaching new location. Also those parameters may be used to perform some additional steps, e.g. notify corresponding nodes about new address. Proper target base station selection may prove to be difficult. During scanning, parameters of available neighbor base stations are detected and the best one is chosen. That base station's ID is sent to current base station as a best candidate for handover. However, due to configuration or other conditions, serving base station may forbid handover to that potential target base station and force subscriber to use another destination. This renders the data gathered in scanning phase obsolete. Such scenario is unlikely, but possible. There are 3 possible approaches to deal with that problem:

(1) *Cooperative*—To initiate handover, Subscriber Station sends list of desired target Base Stations. Assuming

Base Station cooperation, Subscriber will receive permission to execute handover to desired location. In such case, remote DHCPv6 autoconfiguration can be initiated before actual handover procedure is initiated.

(2) *Conservative*—This approach may be considered worst case scenario. Subscriber station assumes that base station will not allow handover to proposed target location, but rather force subscriber station to use different destination. Subscriber station can initiate remote autoconfiguration after BSHO-RSP (IEEE 802.16 message—Base Station initiated Handover Response) message is received from base station. This causes subscriber station to wait for base station response before actual IPv6 preparation can take place.

(3) *Hybrid*—Once scanning is complete, subscriber station has list of possible destination targets. Instead of initiating remote IPv6 configuration for the best target, it starts remote configuration process for all targets, before triggering actual handover. If base station denies request to move to a specified target and provides other location as destination, subscriber station may already have completed configuration retrieval for that location. Subscriber station may then continue with handover process and discard configuration for remaining, not used locations. Theoretically, it is possible that base station may provide destination target that was not previously discovered during scanning procedure, but such behavior is highly unlikely. It would force subscriber station to perform handover to a base station that it was unable to listen to.

As conservative approach is considered the worst case scenario, it was selected for validation during simulation and modeling.

4.1 Network layer independency

Other approaches should be considered as areas for possible further improvements. Proposed method in its current form requires WiMAX as a network layer. This functionality can be easily broadened to other network types, however. It is possible to generalize this mechanism to any networks, but extra mechanism for neighbor discovery is required. MN wants to obtain knowledge about potential handover targets and configuration available at each location.

The most convenient way to obtain such knowledge is to discover and contact neighboring DHCPv6 servers. During initial configuration at current location, MN sends Solicit messages that contain Option Request Option (ORO) with its content specifying options that client would like to have configured. Besides usual options, client also expresses the intent to obtain OPTION_NEIGHBORS option. Server that supports this proposed enhancement will include OPTION_NEIGHBORS in its response. Server must not send

```
 0                   1                   2                   3
 0 1 2-3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-------------------------------+-------------------------------+
|  OPTION_NEIGHBORS (TBD)       |          option-len           |
+-------------------------------+-------------------------------+
|                                                               |
|                1st Neighbor's Server Address                  |
|                        (16 octets)                            |
|                                                               |
+---------------------------------------------------------------+
|                                                               |
|                          . . .                                |
|                                                               |
+---------------------------------------------------------------+
|                                                               |
|                nth Neighbor's Server Address                  |
|                        (16 octets)                            |
|                                                               |
+---------------------------------------------------------------+
```

Fig. 2 Proposed format of the Neighbors DHCPv6 Option. This option could be used to announce possible handover targets, thus eliminating the need to provide this information by network layer

OPTION_NEIGHBORS option to the MN, unless MN explicitly asks for it, using ORO. Proposed format of that option is presented in Fig. 2. It should be noted that such approach to discovery process is backward compatible. Clients that do not support this enhancement will simply not ask for new option. On the other hand, servers that do not support this enhancement and are being asked by clients to send OPTION_NEIGHBORS, will simply ignore the request for unknown option, but will process remaining options normally.

Once knowledge about remote DHCPv6 servers is known, MN may initiate remote autoconfiguration for selected neighboring locations. After choosing one or more suitable targets, MN sends unicast Solicit message to the destination server's address. To notify server that this message is used with intent of remote auto configuration, client should include OPTION_REMOTE_AUTOCONF option. Server responds normally, using Advertise message. Once client receives responses from all remote servers, knowledge about offered parameters at potential destination locations is gained. This information may influence selection of the final target location. Client asks for address and parameters using Request message, again sent to remote server's unicast address. OPTION_REMOTE_AUTOCONF option is also used in the message to indicate that remote mode of operation continues. Server responds with Reply message. Once client receives this message, its remote configuration concludes. Client now possesses all configuration parameters to be used once handover to destination location is complete. Details of this proposal are discussed in [13].

5 DHCPv6 routing configuration

Hosts located in a network configure their routing tables using Router Advertisement (RA) mechanism from Neighor Discovery protocol [20]. It assumes that routers periodically transmit RA messages that are processed by hosts. Alternatively, hosts can explicitly request such announcements by sending Router Solicitation (RS) message. There are limits on maximum frequency of RA transmissions, however. In densely populated network, such limit of 3 seconds (MIN_DELAY_BETWEEN_RAS constant) could be reached easily. Moreover, properly conducted autoconfiguration assumes that host will not initiate DHCPv6 configuration, until RA is received. This results in added delays. As such, it seems reasonable to conduct routing configuration using other means, like DHCPv6.

RA mechanism for configuring routing is affected by number of limitations that affect other use cases, even those not related to mobility. The primary concern is that it is not possible to differentiate between hosts in a network. A simple example is a corporate network that has two routers—one for Internet connectivity and the other for remote site. Most hosts are using only default router. However, selected class of users is also using dedicated router for connections with remote site. It is not possible to configure routing for such scenario. Another limitation is the inability to monitor routing configuration by network operators. Hosts do not send confirmations of any kind, so there is no way to confirm that RA was indeed received. Route selection is also primary concern in multi-homing environments, where hosts receive multiple RAs over multiple interfaces. Default router selection problem is being analyzed by MIF working group of IETF.

To solve aforementioned problems, authors propose new mechanism for routing configuration using DHCPv6. Client sends IA_RT option in its Solicit and Request messages. Server responds with Reply message containing IA_RT option, populated with one or more OPTION_NEXT_HOP options. Each such option represents a single router available in the network. For each OPTION_NEXT_HOP option, there may be one or more OPTION_RT_PREFIX. Each suboption defines dedicated prefix that is available via specified next hop. Proposed solution is designed as a minimal framework for conveying routing information. Its hierarchical approach simplifies future extensions for delivering other routing parameters, like MTU, prefix lifetimes and others. Details of this proposal are discussed in [3]. This proposal was presented during several IETF meetings and gained favorable reviews. It was decided to have it adopted as a MIF work group item.

6 Validation

To support validity of new proposals, it is a common practice to create theoretical models of proposed methods. By studying their properties, conclusions about modeled mechanism can be deducted. Unfortunately, construction of analytical model is only possible for simpler systems. Therefore it is often not feasible to propose reliable model for more complex systems, like multi-subscriber 802.16 networks with advanced IPv6 mechanisms. Commonly accepted approach to mechanism validation is to design and implement a simulation. By running a simulation, various parameters and properties of the simulated proposal can be measured. By analyzing simulation result, one can draw conclusions about properties of the simulated system. It is essential to properly process obtained results as simulation environment and methods may introduce unwanted artifacts and errors to observed values.

6.1 Simulation environment

To verify correctness and evaluate efficiency of proposed mechanisms, simulator of affected systems was developed. OMNeT++ [22] was selected as the environment suitable for that purpose. OMNeT++ is a component-based, modular and open-architecture discrete event network simulator. It allows construction of arbitrary complex networks of interconnected modules. This environment was chosen, due to following reasons:

- *Open source*—source code is available for use and modification. This critical requirement allows modification of any part of the code.
- *Written in C++*—Simulation speed is essential in complicated systems. The scalability of system coded in fast languages (C,C++) are better, compared to slower languages (Java, tcl, perl, etc.)
- *Extensive documentation*—Detailed User's Guide [22] is available, accompanied by extensive set of examples and tutorials.
- *Free*—It is free of charge for personal and academic purposes. OMNeT++ is distributed under Academic Public License.
- *Portable*—OMNeT++ simulation engine runs on Linux, numerous UNIX systems and even Windows. Such broad coverage allows better scalability. Should home PC prove to be not powerful enough to complete calculations, simulation may be run on university cluster. Although simulation efficiency never exceeded modern home PC's capabilities, possibility to use more powerful systems was not ruled out before implementation was complete.
- *Distributed*—OMNeT++ support computation in distributed environment, further increasing scalability.

Table 1 Experiments summary. Most important parameters for each experiment are presented

Experiment	1	2	3	4
# of subscribers	20	7	20	5
Simulation time (s)	60	67	1801	4800
Traffic model	bursty	bursty	bursty	bulk
Packet interval (ms)	12	12	12	11.1
Burst size	3	3	3	1
Packet sizes (min–max)	48–512	48–512	48–512	48–1500
Mobility model	time trigger	time trigger	random time trigger	random time trigger

- *Scalable*—Simple modules may be connected together to form larger, more complex compound modules. This leads to an effective hermetization. As a direct effect, one module may be modified, without any changes to remaining blocks.

Neither OMNeT++, nor any of its available libraries, did not provide support for 802.16 networks simulation, when author began their research. It offers experimental support for IPv6 and Mobile IPv6, but this support relies on precise, but complicated and slow INET framework [21]. Therefore new environment was developed for the purpose of mobile IPv6 stations simulation in the IEEE 802.16 environment. This project was started in 2005 under the name Numbat [12]. Numbat provides means for simulation of 802.16 stations with advanced IPv6 stack on top.

6.2 Traffic models

The main areas of concerns are multimedia applications as they are the most sensitive type of traffic. Therefore most models are related to multimedia streaming or multimedia related purposes. Out of wide variety of available options, following models were implemented (the most important parameters for experiments are presented in Table 1):

- *Bursty traffic*—This traffic model is dedicated to generation of traffic that is variable in time. Once every t_I interval, b_n packets are sent. Packets have random size from L_{min} to L_{max}. The size of packet is a truncated normal distribution, with $L_{avg} = (L_{min} + L_{max})/2$ and standard deviation $\mu = 0.8 * (L_{avg} - L_{min})$. $L_{min} = 48$ [bytes] was selected as lower bound as this is the smallest possible IPv6 packet with useful payload—an empty UDP packet. Example values used in some scenarios are $t_I = 12$ ms, $b_n = 3$ and $L_{max} = 512$ [bytes]. This type of traffic may be used to simulate VoIP connections, where certain amount of data is created in regular intervals;
- *Bulk traffic*—This type of traffic is intended to reflect bulk data transmission, e.g. FTP session or MPEG-2 or H.264 video streaming. It is expected that packet sizes will usually be close to maximum. Smaller packets will also be recorded, albeit on a much smaller scale. Once every tI interval, packet of size L is being sent. There are several distributions that allow modeling such conditions. Beta distribution was chosen as most suitable. It is defined as:

$$f(x;\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

where Γ is defined as:

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$$

Details regarding traffic models are available in [11] or [16].

6.3 Random number generators

Computer is a finite state machine and its next state is fully determined by the previous one. Although true randomness is impossible, number of algorithms has been developed to generate stream of numbers that appear to be random. As they are not truly random, such class is often referred to as pseudo random number generators (PRNG). Although sequences that are closer to truly random can be generated using hardware random number generators, its use is limited by availability of required dedicated hardware. Depending on the expected area of application, there are several properties of a PRNG that should be taken into consideration:

- *Period*. That is considered the most important parameter. All PRNGs generate series of values. Such series are not infinite, but rather repeat themselves. Every time a number is generated, PRNG changes its internal state. Period specifies how many numbers are generated before PRNG returns to its initial state.
- *Weak seeds*. Initial state of the PRNG is defined by a small set of data called seed. Some PRNGs exhibit very limited capabilities for certain seeds (e.g. Middle-Square Method generates only zeros, when 0000 is used as seed.). Existence of weak seeds is considered a serious aw for a PRNG, especially from cryptographic perspective.

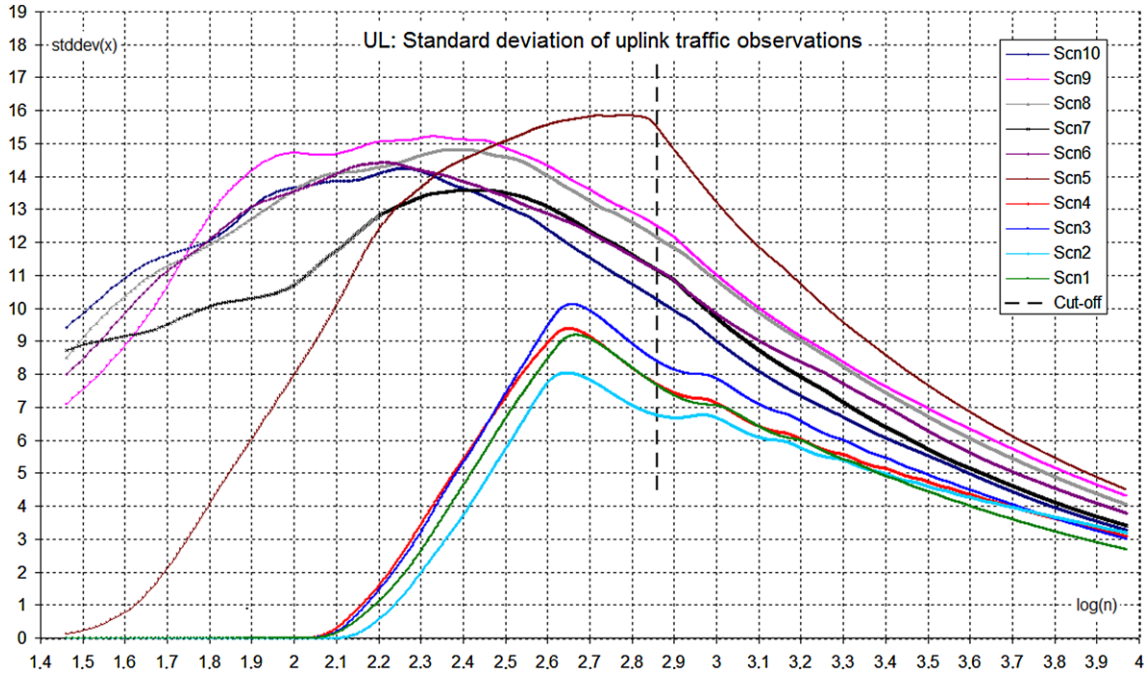

Fig. 3 Standard deviation of uplink traffic observations. It is used for determining initial simulation warm-up time. After stabilization (around $\log(n) = 2.86$, marked with vertical dashed line), standard deviation begins steady decrease with increasing number of samples (n)

- *Computational Complexity*. Some PRNGs require longer computation for next number generation. That parameter is especially important when large quantities of pseudo-random numbers are required. That is particularly true with long lasting simulations.
- *Memory requirements*. As most other algorithms do, also PRNGs require memory to store its internal state. As modern computers have significant amount of memory available, this property is rarely an issue.
- *Warm-up period.* Each PRNG requires initial value called seed that defines initial state. Initially generated sequences of numbers sometimes lack required statistical properties and thus fail random quality tests [7, 8]. Some PRNGs begin to generate high quality number sequences faster than others. Such PRNG are said to be getting started quicker. Such PRNGs are considered more useful.

Mersenne Twister PRNG [9] was found to be suitable and was used in all simulations.

6.4 Simulation warm-up period

The proper determination of simulation warm-up time is one of crucial steps for ensuring simulation credibility. There are numerous methods for choosing length of the warm-up time. Following approach, originally proposed in [19] was adopted and modified. Let there be a sequence of n independent and identically distributed random variables $X_1 :: X_n$, each having finite values of expected value μ and variance $\sigma^2 > 0$. Central limit theorem [2] states that with increasing n, the distribution of the sample average of these random variables approaches normal distribution with a mean μ and variance $\sigma^2 = n$ irrespective of the shape of original distribution. Following linear equation can be obtained:

$$\log s = -0.5 \log n + \log \sigma$$

Therefore, if the simulation approaches steady state, the standard deviation of samples plotted against $\log(n)$ should begin steady decrease. Rate of that decrease should be tangential to line with -0.5 slope. This point defines end of the warm-up period and is often referred to as cut-off point. If fluctuations in samples are high, it may be difficult to spot curve's trend. Therefore moving average algorithm was used to smooth out high frequency fluctuations. For each sample, number of previous and following values was averaged. The moving average of length $2k + 1$ is as follows:

$$\overline{x_n} = \begin{cases} (2k+1)^{-1} \sum_{i=n-k}^{n+k} x_i & \text{if } n \geq k+1 \\ (2k-1)^{-1} \sum_{i=1}^{2n-1} x_i & \text{if } n < k+1 \end{cases}$$

Example of such analysis for uplink traffic is presented in Fig. 3. Reader interested in more detailed explanation is encouraged to read [11].

Table 2 Results summary. Averaged results from all experiments. Values are specified in seconds

Parameter	Scn1	Scn2	Scn3	Scn4	Scn5	Scn6	Scn7	Scn8	Scn9	Scn10
HO Preparation	0,101	0,101	0,103	0,101	0,103	0,103	0,101	0,318	0,319	0,320
DHCP conf. time	2,097	2,113	2,115	2,111	1,079	0,078	0,078	0,223	0,234	0,215
802.16 reentry	0,595	0,081	0,076	0,079	0,082	0,076	0,077	0,088	0,087	0,087
IPv6 conf. time	2,401	2,438	2,463	2,449	1,359	0,315	0,313	0,237	0,242	0,241
Lack of comm.	2,903	2,392	2,394	2,388	1,391	0,400	0,399	0,320	0,315	0,325

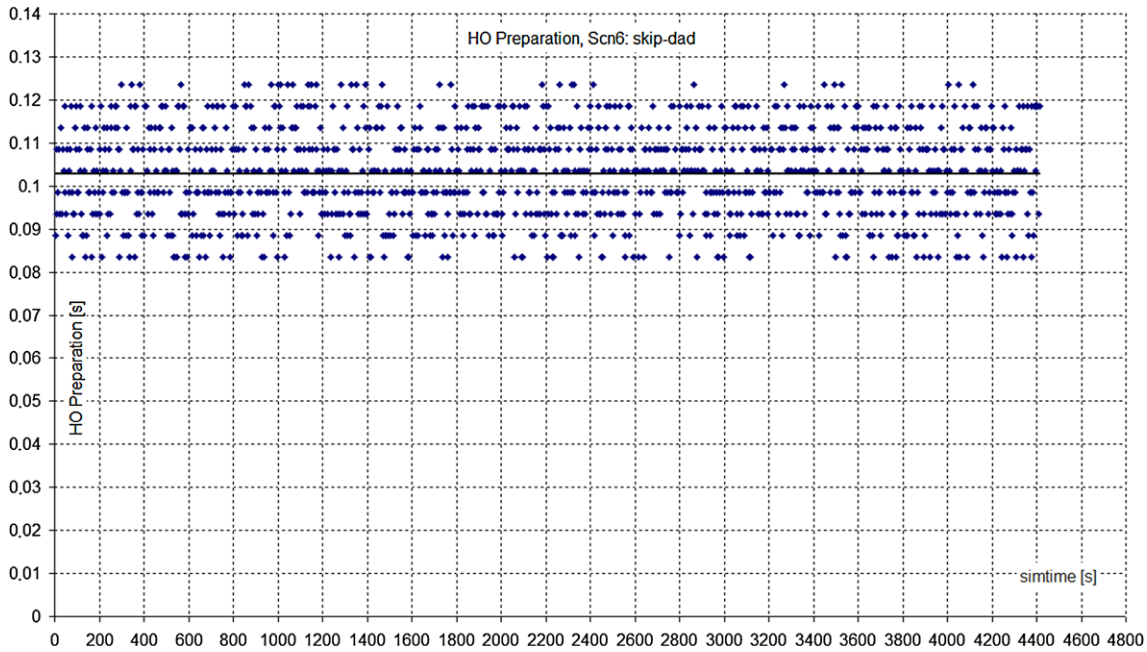

Fig. 4 Handover preparation measurements. Quantization levels of measured HO preparation times are clearly visible

7 Efficiency comparison

Seven different scenarios were assessed during experiments, with each case including additional optimization or mechanism. Scenarios 2 to 5 include optimizations that are allowed by standards (802.16 optimizations, preference 255, skip initial delay and rapid-commit). Scenario 6 assumes that DupAddrDetectTransmits counter [20] is set to 0, effectively disabling DAD procedure. As this breaks Neighbor Discovery specification [20], it is used as a reference scenario only. The main purpose of DAD scenario is not to ignore DAD, but rather assess DAD's impact on handover delays. There are several known ways to improve DAD delays [10]. Scenario 7 introduces server-side DAD [14, 17]. Remote Autoconfiguration is introduced in scenario 8. Scenario 9 introduces routing configured via DHCPv6. Final scenario 10 features example exploitation of knowledge gained via remote autoconfiguration—Binding Update procedure is started before L2 handover is conducted, rather than after arriving at destination location. Summary results of measured parameters for each scenario are presented in Table 2.

Handover preparation time is mostly the same during first 6 scenarios and differs very slightly and there is no improvement in this parameter. Example measurements for handover preparation times are presented in Fig. 4. Unfortunately, with introduction of remote address configuration, handover preparation is slightly increased. It should be noted that HD metric for handover preparation is 0 ms in all cases, as subscriber maintains communication capability. That increase may be considered the necessary cost of shortening other, more critical phases of the handover. If such increase by average 220 ms is deemed too large, there are several options available to address this issue. Handover decision algorithm may be modified to trigger handover earlier. If subscriber's algorithm is not suitable for modification, base station initiated handover may be used instead. Nevertheless, extending preparation phase should not pose any impact on user's experience, as mobile node maintains full connectivity during handover preparation. Also, conservative mode of Remote Autoconfiguration (that is considered the worst case) was simulated. Assuming more optimistic approaches (hybrid or even cooperative), better results may be obtained. Network

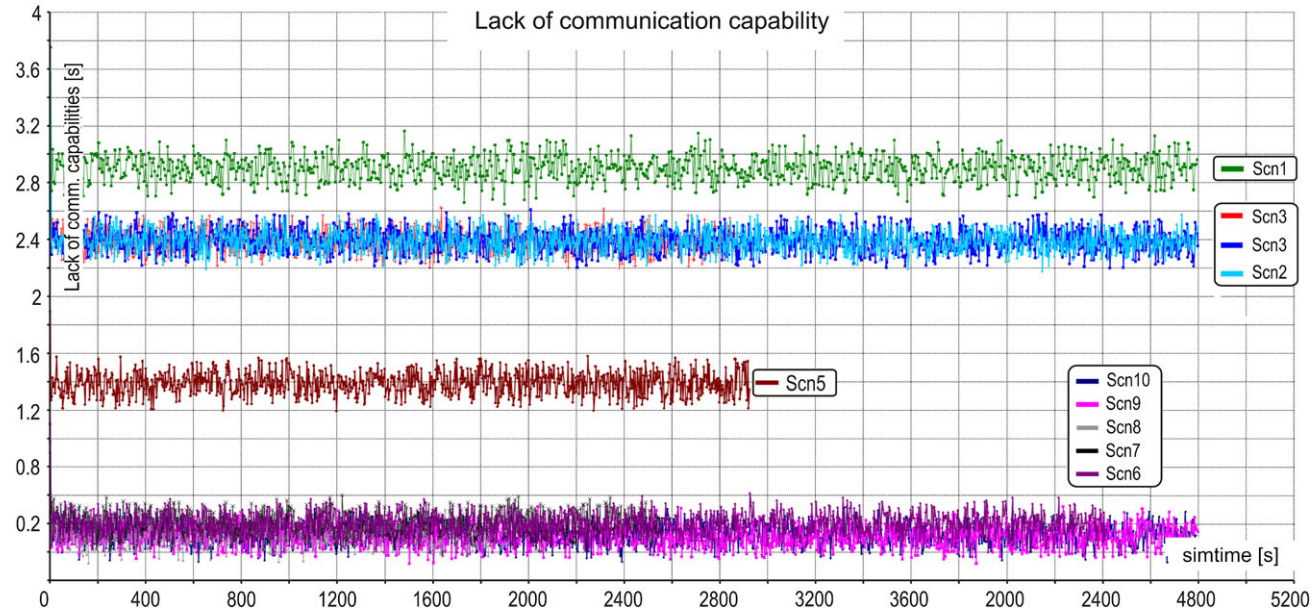

Fig. 5 Lack of communication capabilities. All scenarios can be easily divided into four groups. First group (*green lines*, scenario 1) oscillates around 2.9 s. Second group (scenarios 2 to 4) provides similar blackout period results around 2.4 s. Third group (scenario 5) decrease handover delays further, to 1.4 s. Final group (scenarios 6 and 7) provide even better optimization with delays below 0.4 s

reentry time is only affected by IEEE 802.16 optimizations enabled in scenario 2. Following scenarios (3–10) maintain similar level of roughly 80 ms as all remaining optimizations focus on IP layer rather than 802.16. DHCPv6 configuration time is not affected by any standard improvements (scenarios 2–4), except rapid-commit option introduced in scenario 5. That result can be significantly improved by over an order of magnitude by skipping DAD (scenario 6) or executing DAD on server-side (scenario 7). Remote autoconfiguration (scenario 8) slightly degrades DHCPv6 configuration times, but they are still over 400% better than best standard case.

IPv6 configuration time is part of the IPv6/802.16 handover that takes the most time. Impacted by only one standard mechanism (rapid-commit option, scenario 5), its HD metric varies from 2,449 ms to 1,359 in standard cases. Similar to DHCPv6 configuration time, the bigger improvement is observed with skip DAD related proposals: skip (scenario 6) and server-side (scenario 7). Further improvement on a smaller scale can be achieved thanks to remote autoconfiguration mechanism (scenario 8). Routing configuration via DHCPv6 (scenario 9) and remote Binding Update (scenario 10) have negligible impact on handover delays.

Final and most important parameter—lack of communication capability—may be perceived as a logical (not arithmetic, as some parameters may overlap) sum of all previously investigated parameters. Being the only one that is directly observable by end user, it requires special attention. Being affected by essentially all improvements, it is steadily decreasing with number of enabled mechanisms. Standard based scenarios offer a way to decrease lack of communication capabilities from over 2900 ms in scenario 1 to over 1390 ms in scenario 5, which may be considered a good result. Unfortunately, with HD metric around 1500 ms, the delay is still clearly noticeable by end users. Therefore further improvements are desired. The proposed DAD improvements speed up handover by almost one second, down to 400 ms range. That result can be further improved by enabling remote autoconfiguration (scenario 8 and following), down to 320 ms. Example test measurements from one experiment are presented in Fig. 5. Averaged results are presented in Fig. 6.

8 Conclusion

Using proposed handover latency mitigation techniques, the delay was decreased from 1390 ms to 320 ms. This offers handover delay reduction by over 76%, compared to the best case offered by standards. Several efficiency related mechanisms were analyzed: Skip initial delay in DHCPv6, Skip Duplicate Address Detection, and Remote Autoconfiguration. According to author's knowledge, that is the first proposal related to mobility efficiency in DHCPv6. The idea to perform stateful autoconfiguration remotely, before client is physically attached to the link is new and was never analyzed before. As such, it is a novel solution to a well known problem.

Furthermore, obtained results clearly confirm usefulness of presented proposal. Conducted research allowed major

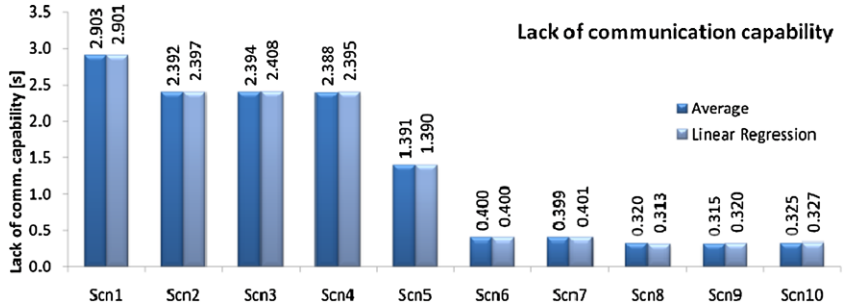

Fig. 6 Comparison of communication capabilities between scenarios. Values specified in seconds

causes of handover latency to be located. Obtained experiment results indicate that the biggest delays are introduced in IPv6 protocol stack. Some parts of IPv6 and DHCPv6 were not designed with mobility in mind. The biggest delay is caused by Duplicate Address Detection in IPv6.

It has been proved that remote autoconfiguration method provides useful way for further shortening of highly optimized handover routines. Thanks to a priori knowledge gained through remote autoconfiguration, mobile node does not waste time on DHCPv6 configuration after changing points of attachment. Validated in several setups, this proposal offers over 20% improvement, even compared to already optimized handovers.

Acknowledgements This work has been partially supported by the Polish Ministry of Science and Higher Education under the European Regional Development Fund, Grant No. POIG.01.01.02-00-045/09-00 Future Internet Engineering. The work has been partially supported by the Polish Ministry of Science and Higher Education under the grant N519 581038.

References

1. 802.21 IEEE working group (2009). *IEEE standard for local and metropolitan networks—Part 21: media independent handover*. New York: IEEE Press.
2. Brandt, S. (1998). *Analiza danych*. Warsaw: Wydawnictwo Naukowe PWN.
3. Dec, W., Mrugalski, T., Sun, T., & Sarikaya, B. (2010). *DHCPv6 route option*. Work in progress, draft-dec-dhcpv6-route-option-05, IETF, Sep. 2010.
4. Droms, R. (Ed.) (2003). *Dynamic host configuration protocol for IPv6 (DHCPv6)*. RFC3315, IETF, July 2003.
5. IEEE 802.16 working group (2006). *802.16e-2004: IEEE standard for local and metropolitan area networks—Part 16: air interface for fixed and mobile broadband wireless access systems*. IEEE Standard, Feb. 2006.
6. Koodli, R. (Ed.) (2009). *Mobile IPv6 fast handovers*. RFC5568, IETF, July 2009.
7. L'Ecuyer, P., & Simard, R. (2009). *TestU01: a software library in ANSI C for empirical testing of random number generators*. http://www.iro.umontreal.ca/~simardr/testu01/tu01.html, April 2009.
8. Marsaglia, G. (1995). *Diehard battery of tests of randomness*. http://www.stat.fsu.edu/pub/diehard/, Florida State University.
9. Matsumoto, M., & Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, *8*(1), 3–30.
10. Moore, N. (2006). *Optimistic duplicate address detection (DAD) for IPv6*. RFC 4429, IETF, Apr. 2006.
11. Mrugalski, T. (2009). *Optimization of the autoconfiguration mechanisms of the mobile stations supporting IPv6 protocol in the IEEE 802.16 environment*. Ph.D dissertation, Gdańsk University of Technology, Gdańsk, Oct. 2009.
12. Mrugalski, T. (2010). *Numbat—mobile IPv6 in WiMax environment*, project homepage. http://klub.com.pl/projects/numbat/, Apr. 2010.
13. Mrugalski, T. (2010). *Remote DHCPv6 autoconfiguration*. Work in progress, draft-mrugalski-remote-dhcpv6-01, IETF, Oct. 2010.
14. Mrugalski, T. (2010). *DHCPv6 server side duplicate address detection*. Work in progress, draft-mrugalski-server-dad-dhcpv6-00, IETF, July 2010.
15. Mrugalski, T., & Woźniak, J. (2008). How to improve the efficiency of IPv6 handovers in IEEE 802.16 networks. In *IEEE Australasian Telecommunication Networks and Applications Conference, ATNAC 2008*, Adelaide, Australia, Dec. 2008.
16. Mrugalski, T., & Woźniak, J. (2009). Analysis of IPv6 handovers in IEEE 802.16 environment. *Telecommunication Systems*, *45*, 191–204.
17. Mrugalski, T., Wozniak, J., & Nowicki, K. (2010). Remote stateful autoconfiguration for mobile IPv6 nodes with server side duplicate address detection. In *Australasian Telecommunication Networks and Applications Conference ATNAC'2010*, Auckland, New Zealand, Nov. 2010. New York: IEEE Press.
18. Soliman, H., Castelluccia, C., ElMalki, K., & Bellier, L. (2008). *Hierarchical mobile IPv6 (HMIPv6) mobility management*. RFC5380, IETF, Oct. 2008.
19. Stankiewicz, R. (2007). *Analytical models of selected DiffServ network elements supporting assured forwarding*. Kraków: AGH University of Science and Technology.
20. Thomson, S., Narten, T., & Jinmei, T. (2007). *IPv6 stateless address autoconfiguration*. RFC 4862, IETF, Sep. 2007.
21. Varga, A. (Ed.) (2009). *INET framework for OMNeT++ 4.0*. http://inet.omnetpp.org/, retrieved on June 2009.
22. Varga, A. (2009). *OMNeT++ user manual, 4.0*. http://www.omnetpp.org, version 3.2, retrieved Jan. 2009.

Tomasz Mrugalski received his M.Sc. degree in computer science from Faculty of Electronics, Telecommunication and Informatics, Gdansk University of Technology (GUT), Poland in 2003. He expects to receive his Ph.D. degree by December 2010. He also works for Intel Corp., where he gained practical experience with experimental WiMAX hardware. His activities are mainly related to DHCPv6, IPv6 and open source software development. He is a member of Steering Committee of Polish IPv6 Task Force and leads several open source projects. The most notable are "*Dibbler*" (open source, portable DHCPv6 implementation) and "*Numbat*" (IPv6/Mobile WiMAX simulation environment). He is also author or co-author of 14 journal and conference papers and participates in IETF activities.

Jozef Wozniak is a Full Professor in the Faculty of Electronics, Telecom-munications and Computer Science at Gdańsk University of Technology. He received his Ph.D. and D.Sc. degrees in Telecom-munications from Gdańsk University of Technology in 1976 and 1991, respectively. He is author or coauthor of more than 200 journal and conference papers. He has also coauthored 4 books on data communications, computer networks and communication protocols. In the past he participated in research and teaching activities at Politecnico di Milano, Vrije Universiteit Brussel and Aalborg University, Denmark. In 2007 he was Visiting Erskine Fellow at the Canterbury University in Christchurch, New Zealand. He has served in technical committees of numerous national and international conferences, chairing or co-chairing several of them. He is a member of IEEE and IFIP, being the vice chair of the WG 6.8 (Wireless Communications Group) IFIP TC6 and the chair of an IEEE Computer Society Chapter (at Gdańsk University of Technology). His current research interests include modeling and performance evaluation of communication systems with the special interest in wireless and mobile networks.

Krzysztof Nowicki received his M.Sc. and Ph.D. degrees in electronics and telecommunications from the Faculty of Electronics at the Gdańsk University of Technology, Poland, in 1979 and 1988, respectively. He is an author or coauthor of more than 100 scientific papers and an author and co-author of five books. His scientific and research interests include network architectures, analysis of communication systems, network security problems, modeling and performance analysis of cable and wireless communication systems, analysis and design of protocols for high speed LANs.